ISO TC184/SC4/WG10 N64

Date: **June 27, 1996** **Supersedes SC4/WG10 N42**

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: n/a **Title:** **The nature of industrial data**

Purpose of this document as it relates to the target document is:

  Primary content
X Issue discussion
  Alternate proposal
  Partial content

Current status: draft
Version 0.7

ABSTRACT:

Many companies and projects are now bringing to SC4 requirements and proposed solutions for co-operative use of Application Protocols and data sharing. Many projects have also investigated potential solutions to these requirements, and have presented their results to the SC4 community. This paper identifies a common approach, based on a synthesis of these results, that may improve and extend the STEP architecture and methodology in order to meet industry's requirements and expectations.

KEYWORDS

STEP
Industrial data
Architecture
Methodology
Generic framework
Core model
Application Protocol
Integration

Document status/dates (dd/mm/yy)

Part Documents

__________ Working draft
__________ Project draft
__________ Released draft
__________ Technically complete
__________ Editorially complete
__________ ISO Committee Draft

Other SC4 Documents

__________ Working
__________ Released
__________ Confirmed
__________ Released

Owner: Matthew West
Address: ISCL/4
Shell International Limited
Shell Centre
London SE1 7NA, UK
Telephone: 44 (0) 171 934 4490
Fax: 44 (0) 171 934 6649
E-mail: 100070.670@compuserve.com

Alternate: Julian Fowler
Address: PDT Solutions
Downs Court, 29 The Downs
Altrincham
Cheshire WA14 2QD, UK
Telephone: 44 (0) 1943 831688
Fax: 44 (0) 1943 831688
E-mail: jfowler@pdtsolutions.co.uk

Comments to Reader:

This revised draft is distributed to members of ISO TC184/SC4/WG10 for review, comment and improvement. This version resolves issues and editorial comments raised against the previous versions (N31 and N42).

Contents

The nature of industrial data

1. Introduction

As an ever widening spectrum of industries have become involved in the development, implementation, and use of standards for industrial data, and have developed expectations of the results and benefits to be gained from the use of the SC4 standards, so a number of fundamental issues have been raised regarding the SC4 architectures and methodologies and their applicability to the broad set of requirements that industry brings to SC4. The most notable of these issues are:

— interoperability of applications, i.e., the ability to communicate information amongst heterogeneous computer systems[1];

— co-operative use of ISO 10303 application protocols;

— co-operative use of the SC4 standards: STEP (ISO 10303), P-LIB (ISO 13584), MANDATE, and Parametrics;

— co-operative use of the SC4 standards with other standards that overlap with the domain of "industrial data", such as SGML and EDIFACT;

— integration of data, i.e., the management of data from diverse sources in an efficient and effective manner;

— development of "data sharing" implementations, i.e., of solutions that combine STEP data models with database management system technologies to facilitate concurrent engineering, life-cycle data management, etc.

This paper proposes a response to these issues that exploits work done in a number of the collaborative projects that contribute to the work of SC4.

1.1 Purpose

This document is intended as a response to actions taken by Matthew West (Shell International Limited, UK) at the ISO TC184/SC4 meetings held in Sydney, Australia (March 1995) and Washington DC, USA (June 1995), to produce a feasibility study for an "AEC Core Model". The interest in the subject

[1] The term "AP Interoperability" has been used to refer to this requirement when the applications support different STEP application protocols. The key issue, however, is that there is an expectation that STEP will facilitate communication amongst different types of application. Although some such requirements may be challenged (particularly those that focus on interoperability based only on shared data structure rather than semantics) SC4 must be able to respond to such requirements. The use of application protocols as standard data specifications for product data communication is almost certainly just part of the solution. The question, then, is how much of the rest of the solution can be standardised, and how much of this standardisation falls within the remit of SC4?

of core models[2] has prompted an expansion in the scope of this document, and a wider survey of the nature of industrial data and its use as it relates to the standardisation work within SC4.

This revised draft is distributed to members of ISO TC184/SC4/WG10 for review, comment and improvement.[3]

1.2 Scope

The scope of this document is the development of standard data specifications that meet industry requirements for data exchange (communication) and data sharing (integration).

1.3 Structure of the document

This document is structured as follows:

— section 1 (this section) describes the purpose, scope and structure of the document;

— section 2 defines key terms and concepts discussed in the document;

— section 3 describes industry requirements for data sharing or integration, and compares and contrasts these with requirements for data exchange or communication;

— section 4 identifies some of the perceived issues against the STEP architecture and methodology;

— section 5 provides a brief overview of some of the projects and initiatives that contribute to standard solutions for (product) data exchange, sharing and management;

— section 6 describes the discovered similarities between these initiatives;

> Previous versions of this document included a section proposing how the common architecture described in section 6 may be used to allow SC4 to fulfil a wider range of requirements. This material has been removed from this document and is being used as the basis for a new paper (“Towards a unified architecture and methodology for industrial data standards”, WG10 N65).

— section 7 presents the conclusions and recommendations of this paper;

[2] In addition to the AEC Core Model feasibility study, the SC4 work programme now includes:

- development of a building & construction core model, as ISO 10303-106;
- an Application Protocol Planning Project (APPP) for engineering analysis, that includes a core model as one of its deliverables;
- active interest in core models within the work of SC4/WG10 “Technical Architecture”;
- at least two application protocols (ISO 10303-221 and ISO 10303-226) that draw on the core model developed by EPISTLE.
- ISO 10303-214, which may be regarded as a core model for the automotive industry.

The work of PDES, Inc. on CDIMs (Context Dependent Information Models) may be classed as a precursor to Core Models

[3] The development of this paper has been funded by Shell International Limited.

— section 8 lists the other documents referenced by this paper.

2. Definitions

NOTE – all definitions given in ISO 10303-1 and ISO 10303-13 are assumed to be valid for this document.

core model: an information model that captures the requirements and knowledge of a number of different domains for a specific industry sector[4].

NOTE - The purpose of developing a core model, within the context of industrial data standards, is to ensure consistency across related standards that use the core model.

EXAMPLES - the Building & Construction Core Model (ISO 10303-106), the POSC EPICENTRE model and the EPISTLE Core Model are examples of core models.

NOTE - the term "integrated industry data model" is used as a synonym for "core model" in this document.

data integration: the process of consolidating data instances from multiple, diverse sources; this process involves mapping between the data models for these sources, and depends on the existence of a common data model that makes such mappings possible.

data sharing: the management of data to provide concurrent creation, access and update of multiple, diverse users and applications of the data.

discipline classification scheme: the set of types that can be used to classify data within a given application or discipline; the elements of a discipline classification scheme are defined and described using the terminology of the domain.

3. Requirements

The "assumed" requirements on STEP are stated in ISO 10303-1:

> *"The objective (of STEP) is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving."*

However, the focus of STEP on data exchange, coupled with the statement quoted above and the development of the standard data access interface (SDAI, ISO 10303-22) has led many to the assumption that STEP *supports* data sharing, rather than (just) acting as a "basis" for it.

[4] This is very much a "first cut" at a definition of core model. The characteristics of core models are discussed in more detail in section 6.3.3.

3.1 Data communication

The development of STEP has focused primarily on requirements for data exchange or communications. Indeed, STEP has been portrayed for many years as the natural successor to CAD/CAM data exchange specifications such as IGES, SET and VDA-FS. Data communication typically requires a tightly constrained data specification that identifies a standard vocabulary and grammar for enterprise transactions. Many different levels of such enterprise transaction have been identified: see, for example, the results of a European workshop on interoperability and integration held in August 1993 [[1]]. This identifies the following different levels of communication or transaction:

i) exchange of data between systems offering similar functionality based on similar internal data representations;

ii) exchange of data between systems offering similar functionality based on different internal data representations[5];

iii) exchange of data between systems offering different functionality;

iv) exchange of data between different enterprise functions:

a) communication between disciplines

b) communication between enterprises

c) communication between different life-cycle phases and activities.

The fourth of these aspects may generally be thought of as being orthogonal to the first three.

The solution offered by STEP to each of these categories of enterprise transaction needs is the application protocol (AP). While the AP development methodology is designed to recognise and fulfil this broad range of requirements, it does lead to the development of many APs, where there are significant overlaps between many of them. One of the drivers of the approach described later in this paper is a belief that “AP interoperability” problems will be reduced if these overlapping APs are designed to be consistent at the requirements level.

This leads to the first recommendation of this paper:

SC4 (and, in particular, working groups 10 and 12) should include within its work programme a more formal analysis of each of these requirements for standardised enterprise transactions.

This analysis should include consideration of the capability of application protocols to meet such diverse requirements without unmanageable proliferation of APs. The benefits of an integrated/harmonised/consistent approach to AP development should also be examined.

[5] This distinction based on internal data representation results from consideration of 3D CAD systems employing different paradigms for the representation of solid models, and the known difficulties of translating between these.

3.2 Data integration

Over the past 2-3 years of SC4 development effort we have observed an increasing level of participation from industry sectors whose primary interest in the SC4 standards is in *life-cycle asset management* (building & construction, process plant, offshore, shipbuilding – i.e., the “AEC” industries). Many of the companies in these industry sectors are faced with the challenge of owning and managing technical data for long periods of time, where the data has originated from many different sources. Such companies *also* require data exchange: however, the balance of current problems, and therefore the greatest perceived benefit to be gained from the SC4 standards, is in the area of data integration.

It is likely, moreover, that as the SC4 standards mature then the industries that have traditionally focused on the data exchange capabilities of STEP, or the exchange of parts libraries using P-LIB, will migrate towards data integration requirements as well. For example, it is easy to imagine that airlines and aviation regulatory authorities would have a key interest in standards-based integration of technical data throughout the life of an aeroplane.

The second aspect of data integration is that which may be required to support concurrent engineering, i.e., shared access to common data by different users, disciplines and applications working on a single project. Functionality to support these needs is currently offered by proprietary “product data management” systems.

Requirements for data integration may be summarised as follows:

— a single logical source for data used by different applications and users;

— management of data from numerous sources;

— management of data across time and life-cycle phases.

3.3 Efficiency

It is a (currently implicit) requirement on the SC4 standards that the resulting software implementations should be efficient. The development of the SC4 standards focuses on the creation and use of conceptual (implementation neutral) data models. Such models are generally characterised as not reflecting efficiency considerations. However, it is practically impossible to ignore such considerations, and they may be used as a basis for choosing between equally "correct" solutions at the conceptual level.

4. Perceived issues and potential solutions

As discussed earlier, STEP’s tantalising mention of “data sharing” has raised expectations that STEP *does* support, or at least enable, data integration. However, consideration of the actual status of STEP is that neither Application Protocols nor Integrated Resources support such needs to the level anticipated by industry:

— the integrated resources do not include many of the application-specific data types required within the logical design for a shared product database[6];

— the application protocols are too constrained (from the enterprise transactions requirement) to be effectively implemented within a database management system[7].

— the requirement for “interoperability” across multiple APs is not fully supported;

Many different solutions have been offered and discussed within SC4. Starting points for such solutions include:

— specialisation and enrichment of the integrated resources (potentially through “migration” of subtypes from AIMs);

— creation of a “union of AIMs” that eliminates invalid or contradictory constraints;

— use of “core models” (which may in some ways be compared to “integrated ARMs”) as the basis for shared data implementations.

This leads to the second recommendation of this paper:

SC4 should consider how the SC4 standards satisfy needs for data integration.

The problem is that STEP currently offers only one route to standardisation – application protocols[8]. Therefore, requirements for data integration are “filtered” through the current capability of STEP.

Alongside the analyses of data communication needs proposed in the previous section, SC4 should give consideration to the needs of data integration. This needs to be coupled with a clear message to national bodies and SC4’s customers in industry: that data integration requirements should not be “disguised” as communication needs, by making every proposal for STEP capabilities a proposal for a new AP.

5. State of the art

Although STEP is now well established as a standard for product data communication, the articulation of requirements for product data *integration* requires an analysis of the potential extension of STEP, and its role as one of the SC4 standards for industrial data, to address such requirements. This analysis falls within the scope of ISO TC184/SC4/WG10, whose terms of reference require it to:

> *“… review, at intervals not exceeding three years, the suitability of the architecture against current progress and future requirements*
>
> *Consideration shall be given to:*

[6] A subsidiary issue is that the integrated resources – indeed, STEP as a whole – are too “product-centric” to be used in the context of enterprise-wide data integration, and have an exchange focus.

[7] Again, there is a subsidiary issue in that multiple APs are perceived as having “conflicting” constraints that lead to ambiguity in the valid populations of “STEP based” databases.

[8] JPF: I have used the analogy that STEP resembles the large, complex hotel complexes in which SC4 tends to meet. The “SC4 hotel” has exactly one entrance, labelled “APs available here”!

> — *other appropriate standards architectures;*
>
> — *existing work within the global research and development community;*
>
> — *prior work in SC4.”*
>
> *ISO TC184/SC4 Resolution 217 (Greenville, October 1994)*

This section therefore identifies and gives a brief overview of some of the more recent initiatives or projects, and compares their solutions to the problems, as a basis for further analysis and discussion.

5.1 STEP

STEP provides a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The implementation of STEP is closely focused on the development of Application Protocols, i.e., standardised data specifications that satisfy identified industrial needs for communication of product data between systems. The architecture and methodology of STEP are described in ISO 10303-13 [[6]].

The semantics of each Application Protocol are specified through the process of interpretation. All Application Protocols undergo the same interpretation process, based upon a single conceptual data model (the Integrated Resources). This method of development leads to a consistent relationship between the Integrated resources and all Application Protocols. It also leads to consistent use of data structures between Application Protocols. Where two or more Application Protocols have common information requirements, Application Interpreted Constructs are identified and used to satisfy the common information requirement. This architectural characteristic results in application systems that can read data produced according to more than one Application Protocol. However, due to varying application contexts, the data may not be useful or meaningful.

5.2 P-LIB

The Parts Library standard (ISO 13584) is designed to define a common structure for neutral and exchangeable libraries which may be implemented in CAD and database systems. The P-LIB approach is based on the ANSI/SPARC three schema architecture. The Application Layer details the concepts needed to permit parts library modelling and exchange, with a Conceptual Model (ISO 13584-10) and a Dictionary Methodology (ISO 13584-42) having been defined. At the logical level, information models are formally specified using EXPRESS. Here General Resources (ISO 13584-20) and a Logical Model (ISO 13584-24) model both the data and its behaviour.

5.3 UN/EDIFACT (Business Information Modelling)

A new framework for developing and documenting EDI messages has recently been put forward by the UN/EDIFACT Joint Rapporteurs’ Business Information Modelling group [2]. The framework is based on a top down approach that converts business world perspectives into formally structured representations of activities and data.

The framework consists of three phases:

— Business Analysis

— EDI Requirements

— EDIFACT Message Design

The Business Analysis phase, defines the business requirements in the particular business area of interest. The requirements are firstly captured using Activity Models that identify the functions and information flows, and Information Models that identify Objects, Entities and their relationships. These are known as the conceptual models.

The EDI Requirements phase further refines the activities in the conceptual models that are to be supported by EDI Message Standards. Again two types of distinct models are used. The EDI Supported Activity model identify the activities that use and generate the data exchanged via messages, the order in which data is generated and used, and the rules that will govern the exchange of the data. The EDI Data Requirements model captures the data and data relationships to be included in the EDI message.

In the third, and final phase, the EDIFACT Message Design Guidelines are applied to translate the EDI Data Requirements Model from the previous phase into existing or new segments, composites, data elements and code values in the EDIFACT directories, and to create the message structures. The EDI Supported Activities model is used to state the purpose and scope of a message, therefore specifying the functionality which computer applications must provide to have consistent implementations of the message.

It should be noted that during each phase, verification processes take place to ensure consistency between the activity models and the data models.

5.4 POSC

The Petrotechnical Open Software Corporation (POSC) has developed, as part of the specification of a software integration platform for the oil exploration and production (E&P) sector, the EPICENTRE data model. EPICENTRE is a large, complex model that covers a number of subject areas within its overall scope, with a focus on subsurface geological information, well logging data, etc. The EPICENTRE model is structured around an underlying framework (“high level model”) that supports analysis in the context of a number of orthogonal subtype trees. The EPICENTRE model is primarily designed as an idealised logical data model supporting interoperability of applications across different E&P databases.

5.5 PISA

The aim of the PISA (ESPRIT 6876) project was to establish an infrastructure for data communication about products and processes. The work focused on three main areas:

— Methodology and Tools

— Operational Platform

— Demonstrations

The PISA project built on previous work on the General AEC Reference Model (GARM) [[7]], and a previous ESPRIT project “IMPPACT” [[8]]. The PISA project developed a methodology and supporting tools to improve the development of information models, their integration, implementation and validation. The project identified six stages of model development, termed the model life cycle:

— Needs analysis - an informal specification of user needs

— Requirements analysis - a formal specification of requirements

— Design specification - a formal, paradigm neutral specification of a solution

— Implementation specification - a formal, paradigm specific specification of an implementation

— Implementation - a system independent implementation

— Executable - an executable system dependent implementation

The needs of the user are documented informally using whatever method suits, for example natural language statements to activity models. These requirements are then formally specified using the PISA developed EXPRESS-R (requirements language), which in turn are transformed into a conceptual schema using the PISA developed EXPRESS-C (conceptual language). Once complete, the conceptual schema is again transformed into specific implementation schemas, which can be used to derive an implementation based on such languages as C, C++, Ada etc. The implementation can then be compiled into an executable for a system dependent implementation. The basis of the PISA conceptual models are a Product and Process Reference Model [[9]], which is based on general principles of layering (levels of abstraction and instantiation), generalisation/specialisation, composition/decomposition, etc.

5.6 MARITIME

The MARITIME project identified that there were four main problems associated with the management of information [3]:

— Timely communication

— Information integration

— Heterogeneous environment

— Assessment and retrieval of information

One of the fundamental aims of the MARITIME project was to develop a life-cycle oriented product model of a ship, as opposed to product models that can only enable “snapshots” of information to be captured. The project firstly identified a need for an activity modelling environment that enabled direct development of data models. Known as the AP Factory, this environment used an extended version of the IDEF0 activity modelling methodology to connect data model entity definitions to the ICOMs (Input/Control/Output/Mechanism) and information flows.

To enable information integration, the project developed “Building Blocks” of EXPRESS entities that represent concepts that are potentially of common interest for more than one domain or application. The Building Blocks can reference and inherit information from other Building Blocks, which together create an abstract data model. Such a methodology enables data models to be developed that contain the same root Building Blocks, therefore leading to the reuse of reference model information. It also allows data models to be both extensible and flexible as new Building Blocks can be added, or old ones replaced without affecting the whole model. By linking the activity model and the Building Blocks, an “interface specification” can be created between two applications.

5.7 EPISTLE

The EPISTLE group have developed a data modelling framework [4] that identifies a set of concepts that are common to many requirements, enabling information to be modelled in the same manner. Using the Framework, they are developing a Core Data Model that will cover the information requirements throughout the life of a Process Plant, enabling the integration of Process Plant information with other enterprise information.

Three fundamental principles shape the Framework:

— Representation of the underlying nature of the things modelled, and not constraints that may not be appropriate under all circumstances.

— A universal context for defining entities, such that models of the same real world things use the same entity types.

— The model covers the complete life cycle, thus capturing the life cycle stages, their relationship to each other, and enabling the handling of different enterprise views.

In the real world, something that exists (real or imaginary) has an associated set of characteristics and associations to other things. The Framework exploits this and has developed four sets of orthogonal subtypes representing different aspects of a thing:

— Subject - the underlying essence of the thing (e.g., material, activity, organisation, characteristic, information content)

— Instantiation - whether it is a class, a specific instance or typical instance

— Life cycle - its life cycle state actual, predicted, required or planned

— Reality - real or imaginary

An entity is defined by choosing a subject subtype, an instantiation subtype, a life cycle subtype and a reality sub type. One of each of the four sets of subtypes is required for the full context of something to be known. Thus a model is built up with entities defined using the Framework. This is a consistent approach to the capture and analysis of requirements and ensures that each entity has the intended universal context. This should enable models that have been developed independently of each other, and that have overlapping domains and scopes, to be integrated.

5.8 Section Summary

Although all the above projects have addressed different problem areas, there are numerous similarities in that they have arrived at similar solutions albeit by different routes. The following is a summary of the findings.

— A structured mechanism to be able to identify the requirements and successfully address the requirements has been developed by most projects.

— The use of activity models to both identify the requirements and act as the functional context for the data usage.

— Need for models that capture the full life cycle information of a product, as well as ‘snapshots’

— The majority of the projects used the ANSI/SPARC three layer architecture, either directly or indirectly.

— Stand alone solutions are not the way forward. Such solutions, unless developed in a shared context, will generally be in conflict and require considerable integration (either at a conceptual or implementation level) before they can be made to work together. Structural and semantic integration, and the *development* of solutions in a shared context, allows flexible and extensible models.

The next section of this document examines the common elements of these initiatives, and discusses an approach to the development of a single architecture and methodology for the SC4 standards in order to meet industry needs.

6. “Archaeological discoveries”

This section presents the results of our analysis of STEP and other relevant initiatives, and the various discoveries made of similarities between them.

6.1 Architectural overview

The projects and initiatives outlined above have much to offer to SC4, allowing the SC4 standards to satisfy a wider range of industry requirements without necessarily requiring significant change to its current architecture, methods, or capabilities.

Figure 1 below illustrates the consistent high level architecture found across these projects and initiatives. In each case we observe:

— a **data specification architecture**, providing publicly defined, extensible data model(s);

— **data specification language(s)**, enabling formal specification of data models;

— a **data access architecture**, including publicly defined implementation specifications;

— a **conformance testing methodology and framework**, that defines how implementations are tested.

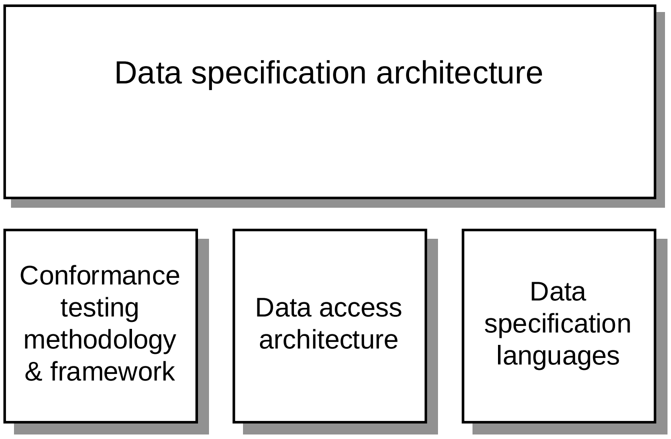
Data specification architecture
Conformance testing methodology & framework
Data access architecture
Data specification languages

Figure 1: high level architecture

Within the SC4 standards, we observe common use of a data specification language (EXPRESS) and implementation methods (physical file, SDAI). However, each of the SC4 standards appears to employ a *different* data specification architecture: this is the root cause of issues related to “co-operative use” of these standards, since it prevents direct re-use of data specifications and therefore sharing of data instances. The conformance testing aspect of the architecture is mature only in the case of STEP and, through its basis in Application Protocols, is closely bound to the data specification architecture.

Based on this high level analysis, the remainder of this paper considers the common aspects found within data specification architectures. Figure 2[9] below illustrates the key common data specification architecture aspects shared by the project results identified above.

[9] This diagram follows the conventions of the CDIF standard. Arrows on the diagram indicate the direction in which to read each relationship; they do *not* indicate any form of dependency.

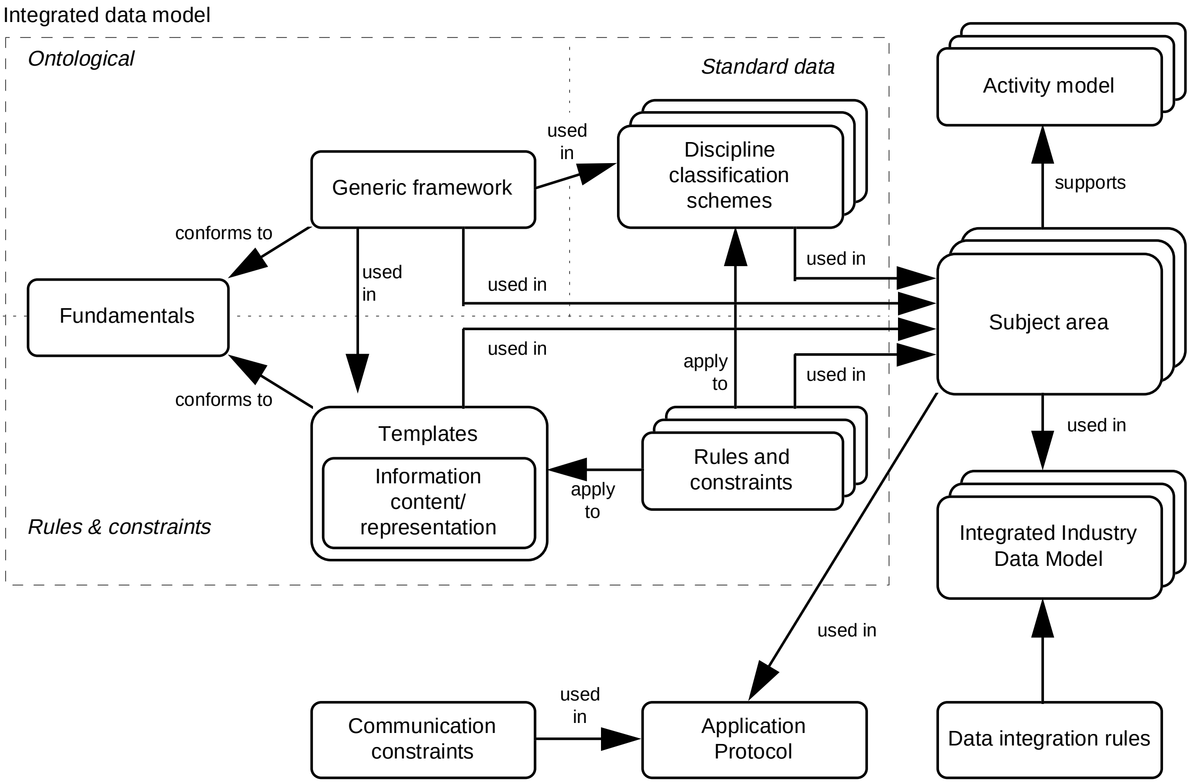

Figure 2: architectural overview

The elements of this architecture may be characterised as follows.

— An integrated data model[10] supporting all industrial data, and uses of that data by an enterprise; this may be further characterised by:

— ontological elements, i.e., the underlying types that are used to construct all more specific models;

— rules and constraints, pre-defined combinations of ontological elements that convey specific semantics;

— standard data that supports the definition of class libraries and discipline specific semantics.

— Subject area models: subsets of the integrated data model for specific purposes[11]. This includes a combination of data content and usage.

— Data models that integrate the subject areas pertinent to a given industry or industry sectors.

[10] Although both STEP and EPISTLE include such an integrated data model, a key distinction is that the STEP integrated data model (the Integrated Resources) is *not* intended to be an implementable model. The EPISTLE model is, by contrast, intended as the basis for shared database implementations (and is being used as such in several industry projects). The reasons for this difference must be understood, and any resulting issues resolved, as part of the improvement and extension process proposed in this paper.

[11] Subject area models are used in EPISTLE, in a similar way that STEP uses Units of Functionality to describe an ARM.

— Exchange protocols that identify the communication constraints that apply within a subject area model for the purpose of exchange.

6.2 Elements of the integrated data model

Integrated data models of this type are already in development or use within projects related to the work of SC4. The POSC EPICENTRE model is an integrated data model for the oil and gas exploration and production industry. Uses of such models include the development of “data warehouse” functions, such as that being undertaken within the ESPRIT “PIPPIN” project.

Figure 1 shows a graphical view of the elements that both make up and use the integrated data model. In this section, each element of the integrated data model is briefly explained and its relationship to elements of the SC4 standards examined.

6.2.1 Fundamentals

Each approach to solutions for product data communication or integration is, inevitably, based on some set of fundamental principles, coupled to a meta-model that determines the common underlying structure of the complete solution. Within STEP, these fundamentals are represented as the generic product data model (GPDM), that is the foundation up on which all the Integrated Resources, and subsequently all AIMs are constructed. Models built upon the GPDM embody the principle of existence dependence, which ensures that all product information is related to an identified product and ultimately to an application context.

Similarly, the EPISTLE approach embodies six development principles [5] for the development of both the integrated data model and subsequent specific data models, ensuring a consistent approach to the capture and analysis of the requirements for data exchange and sharing.

The hierarchical functional requirement and technical solution approach[12] of the GARM is another example of a fundamental principle.

6.2.2 Generic framework

The second common element is the existence of a small, semantically rich data model that forms the basis for extension to cover the complete scope of interest. Such a framework exists in STEP in the form of the “core” schemas of Parts 41 and 43.[13] The results of the General AEC Reference Model (GARM) were a contribution to the STEP integration architecture.

A more formally identified framework is that developed and documented within the EPISTLE group [[4]]. The generic framework is the embodiment of the development principles, and is the set of entity types that will underlie every data model entity type. To this end, the generic framework must cover

[12] The diagrams used to depict the GARM approach have been referred to as “hamburger diagrams”.

[13] JPF: although it is not fully accepted within SC4, I have used here the concept that the GPDM (as a meta model) is separate and distinct from the Integrated Resource schemas that are an instance of the GPDM. Such an approach fits well with “layering” approaches such as those used within the PISA project, i.e., within STEP the IRs are an instance of the GPDM, each AIM is an instance of the IRs, etc.

“life, the universe and everything”. The POSC EPICENTRE High Level Model is another example of this element of the discovered architecture.

6.2.3 Templates

Templates provide a data modelling style and content based upon applying Principles and Generic Frameworks to a particular situation. The STEP Integrated Resources (including the management resources) are templates intended for re-use within Application Interpreted Models. Templates also exist at a higher level of abstraction, i.e., data model structures that form the basis for consistency across a large data model or set of data models. For example, the STEP IRs make consistent use of a template for binary relationships (associations):

```
*)
ENTITY xxx_relationship;
 related_xxx : xxx;
 relating_xxx : xxx;
 name : label;
 description : text;
END_ENTITY;
(*
```

The EPISTLE approach makes similar use of templates, although (as is the case for the generic framework) these templates are documented more explicitly than is the case for STEP.[14] Similarly, the PISA Product and Process Reference Model is primarily concerned with the definition of re-usable templates.

6.2.4 Representation

A common concept in most, if not all, of the initiatives identified above is the distinction between a thing, and information about the thing. The basis of STEP in requirements for CAD/CAM exchange (particularly geometry and drawings) has led to a highly refined mechanism (in fact, another series of templates) for the *definition* of properties (e.g., shape) and the *representation* of shape (e.g., a CAD model). Representation concepts are more fully developed within STEP than any other initiatives examined.

However, the EPISTLE approach embodies a more abstract or generic approach to representation. STEP embodies a single mechanism for representation of product property information; the EPISTLE Generic Framework includes the concept of *information content* and the ability to associate this as a representation of *any* other object (not just properties). This may lead to a higher degree of consistency than is observed in STEP.

EXAMPLE a product has associated information, some of which is available as explicit data and some is given in an identified document. The explicit data pertains to properties such as shape, but also to a company-specific naming scheme. The implicit data also pertains to properties such as material composition.

STEP[15] supports these requirements through:

[14] This difference results from STEP’s existence as a *standard* as opposed to the EPISTLE requirements for “shared work-in-progress”. Some of STEP’s “meta-templates” are, nonetheless, documented as informative annexes to Part 41.

[15] This example is drawn from the interpretation of the requirements of ISO 10303-221. Similar examples may be found in other APs.

- — **product_definition**, **property_definition_representation**, and **representation** for explicit property information;
- — **product_definition.id** for naming;
- — **product_definition_with_associated_documents** for information found in documents.

The EPISTLE approach uses a single concept **information_content** to represent all types of information, and supports a generic association with any other object.

6.2.5 Discipline classification schemes

The existence of fundamentals, frameworks, templates and representation capabilities is not sufficient to capture and fulfil requirements across a broad scope of industry needs. It is also necessary to identify and classify the types of object (and therefore types of data) that are of interest to a particular industry or discipline need.

Within STEP, such classification schemes exist:

- — within the Integrated Resources (particularly the 100-series Application Resources);
- — within AIMs and AICs.

Two mechanisms for the representation of classification schemes are used in STEP:

- — subtyping;
- — constraints on data values.

The latter approach is used in the specification of AIMs; a potential problem with the current approach is that there is no *visible* standardisation of the data values used in different AIMs.

By contrast, the EPISTLE approach makes little use of subtyping for this purpose, relying instead on the definition of “class libraries” that identify standard data used in conjunction with generic templates for classes and classifications.

All sorts of classifications exist in application domains, discipline standards: e.g., IEC standards, building classification codes, etc.

6.2.6 Rules and constraints

The framework, template, and representation elements all enable the definition of very flexible, generic data models. However, in order to meet specific needs associated with usages of the data model it is necessary to place constraints on many of the potential associations that exist within the model. Within STEP, such constraints exist both within the Integrated Resources (primarily as localised rules), and within APs (as mapping rules or local/global rules in an AIM).

6.2.7 Conclusion

The discussion above has demonstrated that the key elements of the “integrated data model” depicted in figure 1 exist in the SC4 standards, and are also common to many of the other initiatives that contribute to the long-term goals of SC4. It should be noted, however, that within STEP some of these elements are localised within Application Protocols rather than being included as part of the integrated data model itself.

In general, the elements of the STEP architecture are somewhat “scattered” throughout the architecture depicted in Figure 2.

6.3 Usage of the integrated data model

This section discusses how the integrated data model described above may be used to achieve data integration.

6.3.1 Functional context: activity models

An activity model identifies the processes that take place in a domain of interest and the information that flows between or act upon those processes. The function of an activity is determined by the combination of Controls acting upon the activity, and the mechanism used to fulfil the activity. Each activity therefore has an associated functional context that is a result of the information flows. The MARITIME project has focused on capturing both the information flows and the context, to enable a life cycle model to be developed. This was achieved by developing an activity model that also captured the information flows as objects in their own right. The MARITIME data model was then developed into an information model by using “building blocks” which had potentially common interest for more than one aspect or domain.

STEP uses Activity Models to identify the domain of interest then the scope of an Application Protocol being developed in that domain, and to discover the data created, modified or transferred within the domain. However, we note that (with some exceptions) Activity Models are still given low priority by many STEP AP projects, to the extent that some are “retrofitted” after the development of the ARM. It is therefore recommended that:

SC4 should encourage much greater attention to the development and use of activity models as a data discovery tool as well as a scoping mechanism.

6.3.2 Subject area models

The subject area models result from partitioning the domain of interest, identified by the activity models, into smaller logical models. As they have a direct relationship to the activity model, they will contain both a data context (what the data model is about), and functional data (how the data is used). As such they have much in common with AICs within STEP. However, there are a number of key distinctions:

— The subject area models are based upon the integrated data model and thus to an extent are already “pre integrated”;

— As more subject area models are developed, there is a requirement for an integration process to ensure that there is no redundancy across subject area models, leading to the extension and enrichment of the discipline classification schemes within the integrated model.[16]

— The subject area models are not focused purely on data exchange requirements.

6.3.3 Integrated industry models

An integrated industry data model is the *union*[17] of the subject area models required to manage the broad industrial purpose such as managing a process plant throughout its life. The already integrated data models can then be associated with each other, thereby producing a set of core model integration rules. This paves the way for the core model to have a dual role, firstly as the integrator of the subject models (ensuring no redundancy), and secondly as a logical model that is suitable for the basis of a design of a database.

This data model should be suitable as the basis for a database design. This is because the subject area models are themselves consistent, and that suitable rules regarding the integration of the subject models have been developed. The first is true as all the subject models have been developed using the integrated data models. The second will be true if the associations between the subject area models are maintained using the generic framework and templates.

ISO 10303-214 (Automotive), the EPISTLE Core Model (Process Plant), and the POSC EPICENTRE model (E&P) are all examples of Integrated Industry Data Models.

6.3.4 Integration rules

Integration rules are required at the instance level to ensure that the integrity of the population of a model can be preserved.

6.3.5 Communication constraints

The requirements of data communication introduce constraints into the models used to achieve communication. For example, the STEP use of the principle of existence dependence enforces completeness of concept that is necessary for communication but may not be applicable for data integration.

6.3.6 Exchange protocols

The exchange protocols are based upon the subject area models. In isolation, they can be used for data exchange, but used in conjunction with the core model, they can be used as the basis for data sharing

[16] It should be noted that this approach differs to STEP, as the integrated resources have been developed so as not to allow the migration of discipline dependent data into the core. However the STEP methodology does have three processes that are analogous. Firstly integrated resources can be extended in response to missing or novel application needs. Secondly. an application resource can be developed, and finally Application Interpreted Constructs (AICs) can be created

[17] In the sense of a set operation, i.e., the elements of each subject area model occur once and only once in the integrated data model.

using a shared data base. Communication constraints may be placed upon an Exchange Protocol to aid efficiency when using different communication techniques.

7. Conclusions and recommendations

This paper has discussed a number of common requirements for extensions to the current data communication capabilities of STEP, and identified a number of potential sources of solutions to aspects of these requirements. Analysis of these solutions against the current SC4 architectures and methodologies suggests two possible ways forward:

— initiation of a work programme to investigate and develop a separate (set of) standard(s) for data integration, compatible with the current SC4 standards;

— a controlled, incremental improvement to the current SC4 architectures and methodologies that enables SC4 to fulfil a wider range of industry needs.

Any changes or enhancements to the SC4 development methodology must also increase the efficiency and reduce the cost of current standards development. However, the detailed implications on the structure and content of the STEP Integrated Resources has not yet been assessed, and specific changes to the modelling methods and practices of Integration and Interpretation have not been identified.

The following recommendations have been identified:

— SC4 (and, in particular, working groups 10 and 12) should include within its work programme a more formal analysis of each of these requirements for standardised enterprise transactions.

— SC4 should consider how the SC4 standards satisfy needs for data integration.

— SC4 should encourage much greater attention to the development and use of activity models as a data discovery tool as well as a scoping mechanism.

In addition:

— SC4 should consider the following data integration solutions and identify possible routes for their exploitation.

 — expansion of the scope of the STEP Integrated Resources beyond “product” data;

 — specialisation and enrichment of the integrated resources (potentially through “migration” of subtypes from AIMs); this should include incorporation of changes resulting from issues arising from the development and implementation of the “initial release” of STEP[18];

 — creation of a “union of AIMs” that eliminates invalid or contradictory constraints;

 — use of “core models” (which may in some ways be compared to “integrated ARMs”) as the basis for shared data implementations.

[18] A similar conclusion has been reached by WG10 within its discussions of the AP Interoperability issue.

— WG10 (with other WGs as appropriate) should conduct a detailed review of this document and the work referenced by it in order to produce a recommendation to SC4 with respect to fulfilment of industry needs for data sharing and integration. Resources permitting, the results of this review should be available to SC4 at the Toronto (October 1996) meeting.

We note with concern that this last recommendation was made in the previous version of this document in the context of the Kobe (June 1996) meeting. WG10 members are encouraged to make use of the WG10 email exploder to comment on this document and use such discussion as the basis for an agreed recommendation to be presented at the Toronto meeting.

8. References

[1] Geilingh, W. F., and Goult, R. ‘Requirements for model integration and interoperability’. ISO TC184/SC4/WG5 N133, October 3, 1993.

[2] Clark, C., and Files, D. ‘The Business and Information Modelling Framework for UN/EDIFACT’. UN/EDIFACT BIM2-811, April 26, 1995.

[3] Lehne, M., and Mehta, S. ‘The History and Future Outlook of PDT within Shipbuilding’. European Product Data Technology Days - Using STEP in Industry, Conference Proceedings, 25-26th January 1995, Munich, Germany.

[4] Angus, C. ‘EPISTLE Framework V2.0’. EPISTLE Issue 1.00, 19th September 1995.

[5] West, M. ‘Developing High Quality Data Models Volume 1: Principles and Techniques’. Shell International Petroleum Company Limited, Report Number IC94-033, March 1994.

[6] ISO 10303-13 “Development methods: Architecture and methodology reference manual”. ISO TC184/SC4/WG10 N30, October 1995.

[7] General AEC Reference Manual. *Reference to be completed.*

[8] Gielingh., W. F. and Suhm, A. (eds). IMPPACT Reference Model. An Approach to Integrated Product and Process Modelling for Discrete Parts Manufacturing. Springer-Verlag, Berlin, 1993

[9] PISA book (draft). ESPRIT project 6876 document D3304, March 1995.